Installing and Using the Microsoft IE Web Controls

- Installing the Microsoft IE Web Controls
  1. Download the installation file from: http://www.asp.net/IEWebControls/Download.aspx?tabindex=0&tabid=1
  2. Make sure you have installed the .NET Framework SDK v1.0 or v1.1.
  3. Modify Build.bat file to the follwoing:

```
---------------------------------------------------------------------------------------------------------------
@if "%_echo%"=="" echo off

if not exist build mkdir build

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\csc.exe /out:build\Microsoft.Web.UI.WebControls.dll @IEWebControls.rsp
xcopy src\Runtime build\Runtime /E /Y /I /Q
---------------------------------------------------------------------------------------------------------------
```

  4. Then follow the readme.txt file comes with the installation file.

- Using the Microsoft IE Web Controls

  Adding the Web Controls to the toolbox in Visual Studio .NET
  1. Start Visual Studio .NET
  2. Open the toolbox, right click it, and select Customize Toolbox.
  3. Select the ".NET Framework Components" tab.
  4. Click on the "Browse" button
  5. Navigate to the folder in which the DLL is installed. By default, the installation folder is "C:\Ptram Files\Microsoft Internet Explorer WebControls".
  6. Select the " Microsoft.Web.UI.Webcontrols.dll " file.
  7. Click "Open" to return to the Customize Toolbox dialog.
  8. Check the boxes for the IE Web controls to add to the Toolbox. The IE Web Controls are MultiPage, TabStrip, Toolbar, and TreeView. Their namespace is Microsoft.Web.UI.WebControls.
  9. Click OK to accept the changes.
  10. Exit and then restart VS.NET to save these changes.

  Another way to add the Web Controls to the toolbox is:
  1. Open the project you want to use the IE Web controls.
  2. From the solution explorer, right click " Reference" to add reference.
  3. Select the "Project" tab, then click the "Browse" button, navigate to the folder in which the DLL is installed.
  4. Select the " Microsoft.Web.UI.webcontrols.dll " file, then click "Open".

  Using the IE Web Controls in the web application

In order to make the web server recognize the web controls, you need to copy webctrl_client\1_0\ directory to your web server's \Inetpub\wwwroot directory.

- Modifying the Microsoft IE Web Controls - TreeView
  The problem with the original Web TreeView is that it does the page "Post Back " when you just expand or collapse the tree. We want to modify it's behavior so that the "AutoPostBack" property can be separated to "AutoPostBackOnCheck" , "AutoPostBackOnSelectedIndexChange", "AutoPostBackOnExpand" and "AutoPostBackOnCollapse".

  - Find the treeview.cs from the \src directory of the IE web Controls.
  - Comment out the "AutoPostBack" property.
  - Add the following code:

```
// ************************************************************
//  The following was added by Ying Tang in order to add properties to the tree   7/27/05
//************************************************************

            /// <summary>Whether or not to post back to the server when expand the tree on each interaction</summary>
            [
            Category("Behavior"),
            DefaultValue(false),
            PersistenceMode(PersistenceMode.Attribute),
            ResDescription("AutoPostBackOnExpand"),
            ]

            public bool AutoPostBackOnExpand
            {
                  get
                  {
                        object b = ViewState["AutoPostBackOnExpand"];
                        return ((b ==null) ? false: (bool)b);
                  }
                  set
                  {
                        ViewState["AutoPostBackOnExpand"] = value;
                  }
            }

            /// <summary>Whether or not to post back to the server when collapse the tree on each interaction</summary>
            [
            Category("Behavior"),
```

```
DefaultValue(false),
PersistenceMode(PersistenceMode.Attribute),
ResDescription("AutoPostBackOnCollapse"),
]
public bool AutoPostBackOnCollapse
{
      get
      {
            object b = ViewState["AutoPostBackOnCollapse"];
            return ((b == null) ? false: (bool)b);
      }
      set
      {
            ViewState["AutoPostBackOnCollapse"] = value;
      }
}

/// <summary>Whether or not to post back to the server when selected node changed on each interaction</summary>
[
Category("Behavior"),
DefaultValue(false),
PersistenceMode(PersistenceMode.Attribute),
ResDescription("AutoPostBackOnselectedIndexChange"),
]
public bool AutoPostBackOnselectedIndexChange
{
      get
      {
            object b = ViewState["AutoPostBackOnselectedIndexChange"];
            return (( b == null) ? false : (bool)b);
      }
      set
      {
            ViewState["AutoPostBackOnselectedIndexChange"] = value;
      }
}

/// <summary>Whether or not to post back to the server when check a node on each interaction</summary>
[
Category("Behavior"),
DefaultValue(false),
PersistenceMode(PersistenceMode.Attribute),
ResDescription("AutoPostBackOnCheck"),
```

```
            ]

            public bool AutoPostBackOnCheck
            {
                  get
                  {
                        object b= ViewState["AutoPostBackOnCheck"];
                        return (( b == null) ? false : (bool) b );
                  }
                  set
                  {
                        ViewState["AutoPostBackOnCheck"] = value;
                  }
            }

//  **********  end of codes added by Ying Tang ****************
```

This will add the appropriate extra properties.

- Replace code in the procedure “RenderUpLevelPath” with the following

```
// ****************************************************************
//   The following code was added by Ying Tang in order to make the new properties work – 7/27/05
//  ***********************************************

if (Page != null)
{
      string strOnExpand = "javascript:" + " if (this.clickedNodeIndex != null) this.queueEvent('onexpand', this.clickedNodeIndex);";
      string strOnCollapse = "javascript:" + " if (this.clickedNodeIndex != null) this.queueEvent('oncollapse', this.clickedNodeIndex);";
      string strOnCheck = "javascript:" + " if (this.clickedNodeIndex != null) this.queueEvent('oncheck', this.clickedNodeIndex);";
      string strOnSelectedIndexChange = "javascript:" + " if (event.oldTreeNodeIndex != event.newTreeNodeIndex)
this.queueEvent('onselectedindexchange', event.oldTreeNodeIndex + ',' + event.newTreeNodeIndex);";
      if (this.AutoPostBackOnExpand == true) {strOnExpand = strOnExpand + Page.GetPostBackEventReference(this, "");}
      if (this.AutoPostBackOnCollapse == true) {strOnCollapse = strOnCollapse + Page.GetPostBackEventReference(this, "");}
      if (this.AutoPostBackOnCheck == true) {strOnCheck = strOnCheck + Page.GetPostBackEventReference(this, "");}
```

```
      if (this.AutoPostBackOnselectedIndexChange == true) {strOnSelectedIndexChange = strOnSelectedIndexChange +
Page.GetPostBackEventReference(this, "");}
      output.AddAttribute("onexpand", strOnExpand);
      output.AddAttribute("oncollapse", strOnCollapse);
      output.AddAttribute("oncheck", strOnCheck );
      output.AddAttribute("onselectedindexchange", strOnSelectedIndexChange);
}

// ********************************************************
//   end of the codes added by Ying Tang
//********************************************************
```

Note that the Microsoft does not support IE Web Controls now. But they do host a newsgroup and a forum. The forum's web site is:
http://forums.asp.net/91/ShowForum.aspx

- Moving Web Project from Local Host to a Web Server

1. Create a virtual directory on the web server.
2. Copy all the files from your local host project directory to the corresponding web server directory.
3. In order to let the Asp.Net web server talk to the SQL2000 server directly, specify the user name and password for the anonymous access for the web server. The specified user name should be name recognizable by the SQL 2000 server.
4. Change the web.config file for this web application. The following needs to be included.
5.
```
<authentication mode="Windows" />
<identity impersonate="true" />
```

6. Copy webctrl_client\1_0\ directory from the local host to the web server's \Inetpub\wwwroot directory.